

NASA/WVU Software IV & V Facility
Software Research Laboratory
Technical Report Series

NASA-IVV-96-017
WVU-SRL-96-017
WVU-SCS-TR-96-28

*IN-61-CR
008186*

V&V Within Reuse-Based Software Engineering

A Position Statement for "A NASA Focus on Software Reuse"

by Edward A. Addy

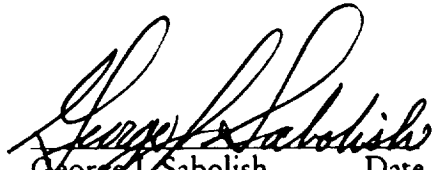


National Aeronautics and Space Administration



West Virginia University

According to the terms of Cooperative Agreement #NCCW-0040,
the following approval is granted for distribution of this technical
report outside the NASA/WVU Software Research Laboratory


George J. Sabolish Date
Manager, Software Engineering


John R. Callahan Date
WVU Principal Investigator

V&V WITHIN REUSE-BASED SOFTWARE ENGINEERING

A Position Statement for "A NASA Focus on Software Reuse"

Edward A. Addy

NASA/WVU Software Research Laboratory

Introduction

Verification and Validation (V&V) is used to increase the level of assurance of critical software, particularly that of safety-critical and mission-critical software. V&V is a systems engineering discipline that evaluates the software in a systems context, and is currently applied during the development of a specific application system. In order to bring the effectiveness of V&V to bear within reuse-based software engineering, V&V must be incorporated within the domain engineering process.

Verification and Validation in Traditional System Application Engineering

V&V has been performed during application system development, within the context of many different development methodologies. V&V is a set of activities performed in parallel with system development and designed to provide assurance that a software system meets the operational needs of the user. It ensures that the requirements for the system are correct, complete, and consistent, and that the life-cycle products correctly implement system requirements. The V&V process evaluates software in a systems context, using a structured approach to analyze and test the software against system functions and against hardware, user and other software interfaces.

A set of V&V activities is defined in the IEEE Standard for Software Verification and Validation Plans, IEEE STD 1012. These

activities are divided into the life-cycle phases listed below.

- Management of V&V
- Concept Phase V&V
- Requirements Phase V&V
- Design Phase V&V
- Implementation Phase V&V
- Test Phase V&V
- Installation and Checkout Phase V&V
- Operations and Maintenance Phase V&V

Verification and Validation in Reuse-Based Software Engineering

One model for reuse-based software engineering is the STARS Two Life-Cycle Model, shown in Figure 1. This model assumes a domain-specific, architecture-centered approach to software reuse.

A working group at Reuse '96 created a framework for V&V within reuse-based software engineering by adding V&V activities to the STARS Two Life-Cycle Model. The group also considered how the new domain-level and transition-level tasks would impact the scope and level of the traditional application-level tasks. The resultant model is shown in Figure 2.

Domain-level V&V tasks are performed to ensure that domain products fulfill the requirements established during earlier phases of domain engineering. Transition-level tasks provide assurance that an application artifact correctly implements the corresponding domain artifact. Application-

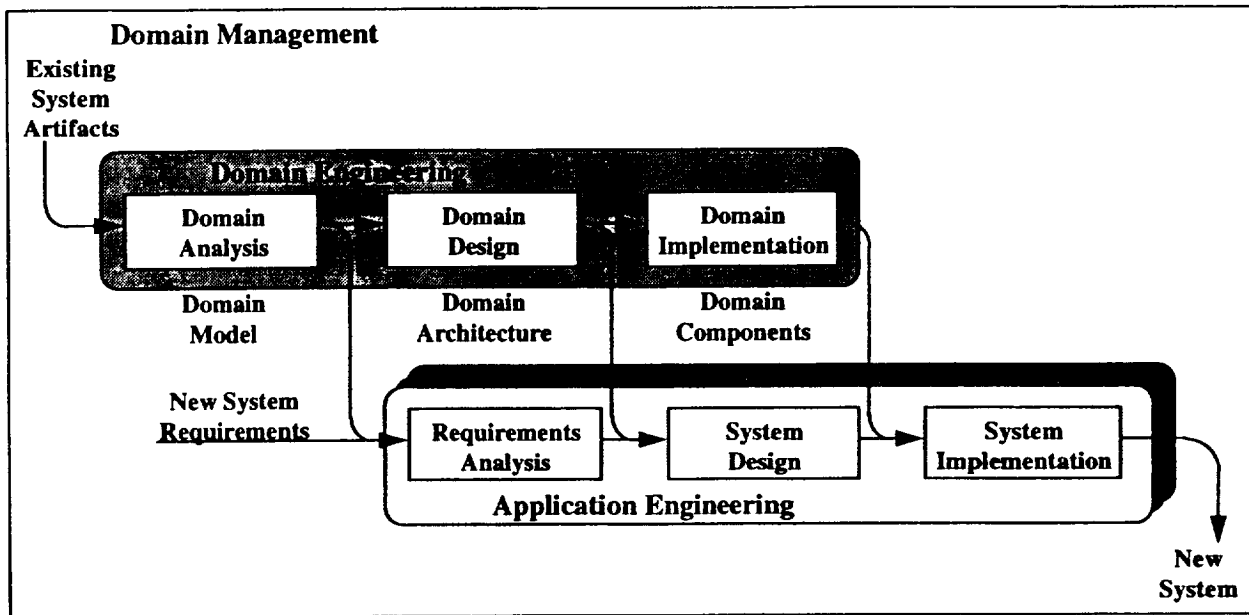


Figure 1: STARS Two Life-Cycle Model

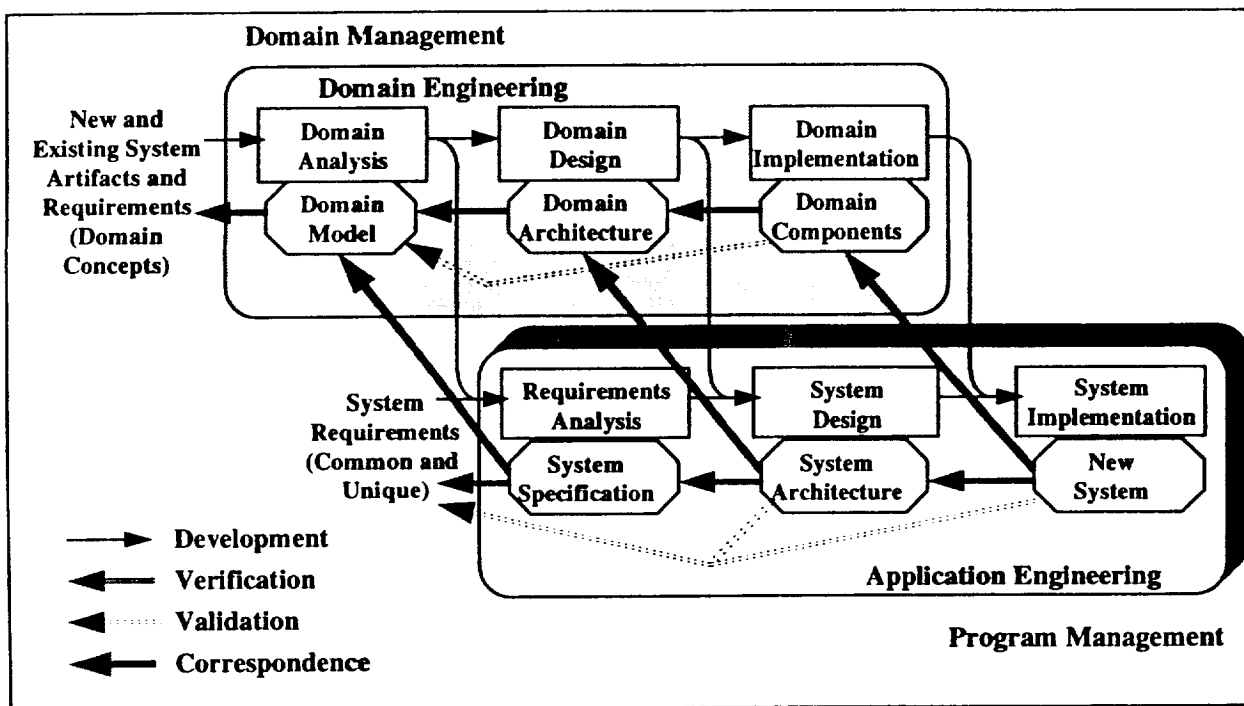


Figure 2: Framework for V&V within Reuse-Based Software Engineering

level V&V tasks ensure the application products fulfill the requirements established during previous life-cycle phases.

The model uses the term “correspondence analysis” to describe the activities to provide assurance that an application artifact is a correct implementation of the domain artifact. Four activities are to be performed during correspondence analysis:

- Map the application artifact to the corresponding domain artifact.
- Ensure that the application artifact has not been modified from the domain artifact without proper documentation.
- Ensure that the application artifact is a correct instantiation of the domain artifact.
- Obtain information on testing and analysis on a domain artifact to aid in V&V planning for the application artifact.

No application-level V&V tasks should be eliminated due to tasks being performed at the domain or transition levels. It might be possible to reduce the level of effort for some application-level tasks, in a case where the application artifact is used in an unmodified form from the domain component, or where the application artifact is an instantiation of the domain component through parameter resolution.

Domain maintenance and evolution should be handled in a manner similar to that described in the operations and maintenance phase of application-level V&V. Changes proposed to domain artifacts should be assessed by V&V to determine the impact of the proposed correction or enhancement. If the assessment determines that the change will impact a critical area or function within the domain, appropriate V&V activities should be repeated to assure the correct implementation of the change.

Communication of the V&V work products and results is vital to avoid the repetition of V&V tasks and to ensure that potential reusers could properly assess the status of reusable components. V&V work products and results should be associated with the component and made available to domain and application engineers. In some cases, the V&V might be directed at a grouping of components rather than at an individual component, and this information should also be available. The information that should be communicated should include the following:

- V&V Planning Decisions and Rationale
- V&V Analysis Activities
- V&V Test Cases and Procedures
- V&V Results and Findings

Conclusion

The primary motivation for V&V at the domain level is to provide assurance that the domain requirements are correct and that the domain artifacts correctly implement the domain requirements. A secondary motivation is the possible elimination of redundant V&V activities at the application level. Domain-level V&V activities should be considered within a domain-specific, architecture-centered domain that contains multiple critical systems.

Future work that needs to be performed to continue development of the framework for performing V&V within reuse-based software engineering includes determining criteria for identifying domains where V&V is appropriate; specifying prerequisites, inputs and outputs for the domain-level and transition-level V&V tasks; and developing methods and tools to perform the domain engineering V&V tasks. Refinement of the framework will occur when experiments are conducted in applying V&V within critical domains.